Mathematics

HP COMPUTER CURRICULUM

Number Sets

TEACHERS ADVISOR

HEWLETT PACKARD

HEWLETT hp PACKARD

Hewlett-Packard
Computer Curriculum Series

mathematics
TEACHER'S ADVISOR

number sets

by E. A. Herber
Golden High School
Golden, Colorado

edited by Jean Danver and Christine Doerr
Hewlett-Packard

Hewlett-Packard Company
11000 Wolfe Road
Cupertino, California 95014

HP 5951-4442

March 1973

The Hewlett-Packard Computer Curriculum Series represents the published results of a Curriculum Development project sponsored by the Data Systems Division of Hewlett-Packard Company. This project is under the directorship of Jean H. Danver.

This material is designed to be used with any Hewlett-Packard system with the BASIC programming language such as the 9830A Educational basic, and the 2000 and 3000 series systems.

The Hewlett-Packard Company makes no warranty expressed or implied, and assumes no responsibility in connection with the operation of the program material contained herein.

TABLE OF CONTENTS

INTRODUCTION

This Mathematics Set of the Hewlett-Packard Computer Curriculum Series consists of a set of a Student Lab Book and a corresponding Teacher's Advisor. It was designed to help meet the need for computer-oriented problems in mathematics, providing students an opportunity to use a computer as a problem solving tool or a modeling device within a particular subject matter area.

The materials are designed for flexible use as desired by the individual instructor. The material and exercises in this unit are intended as an "enrichment" experience in the field of number sets. The unit begins with a section on integers, with exercises involving common operations which are easily performed with a computer. The Euclidean Algorithm and the "Sieve of Eratosthenes" are presented as alternatives to exhaustive search routines for finding greatest common divisors and prime numbers, respectively. The second section deals with rational numbers, with exercises involving denumerability and forming rational numbers from given integer sets. The final section introduces irrational numbers; in particular, methods are derived for computing π and finding the rth root of a number. Obviously, the unit is not intended to be a complete study of number sets; instead, each discussion is self-contained, so the concepts can be studied independently and in any sequence. Thus, the material can be used to supplement and enrich your curriculum in any fashion you choose.

The mathematical concepts needed for each exercise are briefly reviewed, but you may want your students to study these in greater detail before attempting the exercises, especially if their background in number sets is limited. A list of possible references is included at the end of each section. These references will also provide additional problems for your better students.

The degree of difficulty of the material is dependent upon the amount of assistance which you choose to provide. With no assistance, the better student should be challenged. However, given a good deal of assistance, any second year algebra student should be able to work out the exercises with no great difficulty. The level of the material is determined by the assumption that students taking second year algebra will be quite capable as a group.

The Student Book provides text material and programming exercises for the students. There is a problem analysis, including a suggested approach and a macro flow chart, for each exercise. The Teacher's Advisor contains an example of a program to solve each exercise, micro flow charts, and a brief discussion of the important elements of the exercise. The micro flow charts should be given to the students only after they have made an attempt to solve the problem on their own. For best results, you should study all the solutions until you are certain you have a complete grasp of the general methods. This should be done before assigning the material to the class.

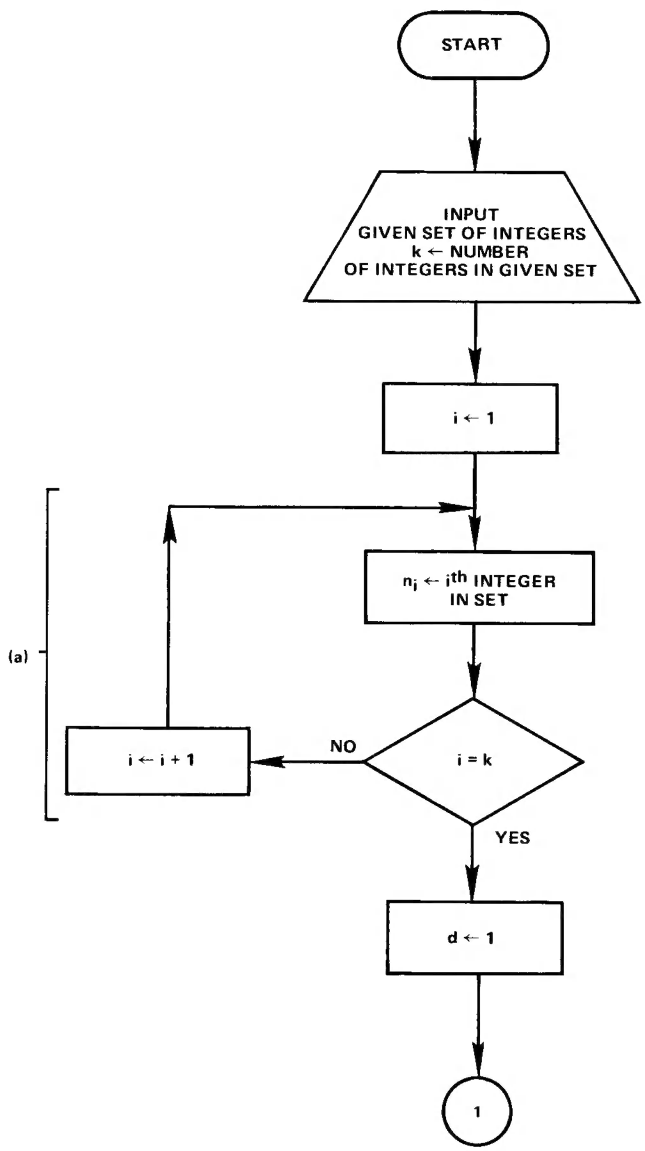
You will undoubtedly think of different programming methods or techniques as you study the example programs. Encourage the students to do the same. There are no *approved* solutions. All solutions are acceptable if they produce the correct results. At this level, there is no need for emphasis on the efficiency of a student's program. The important question is, does it work?

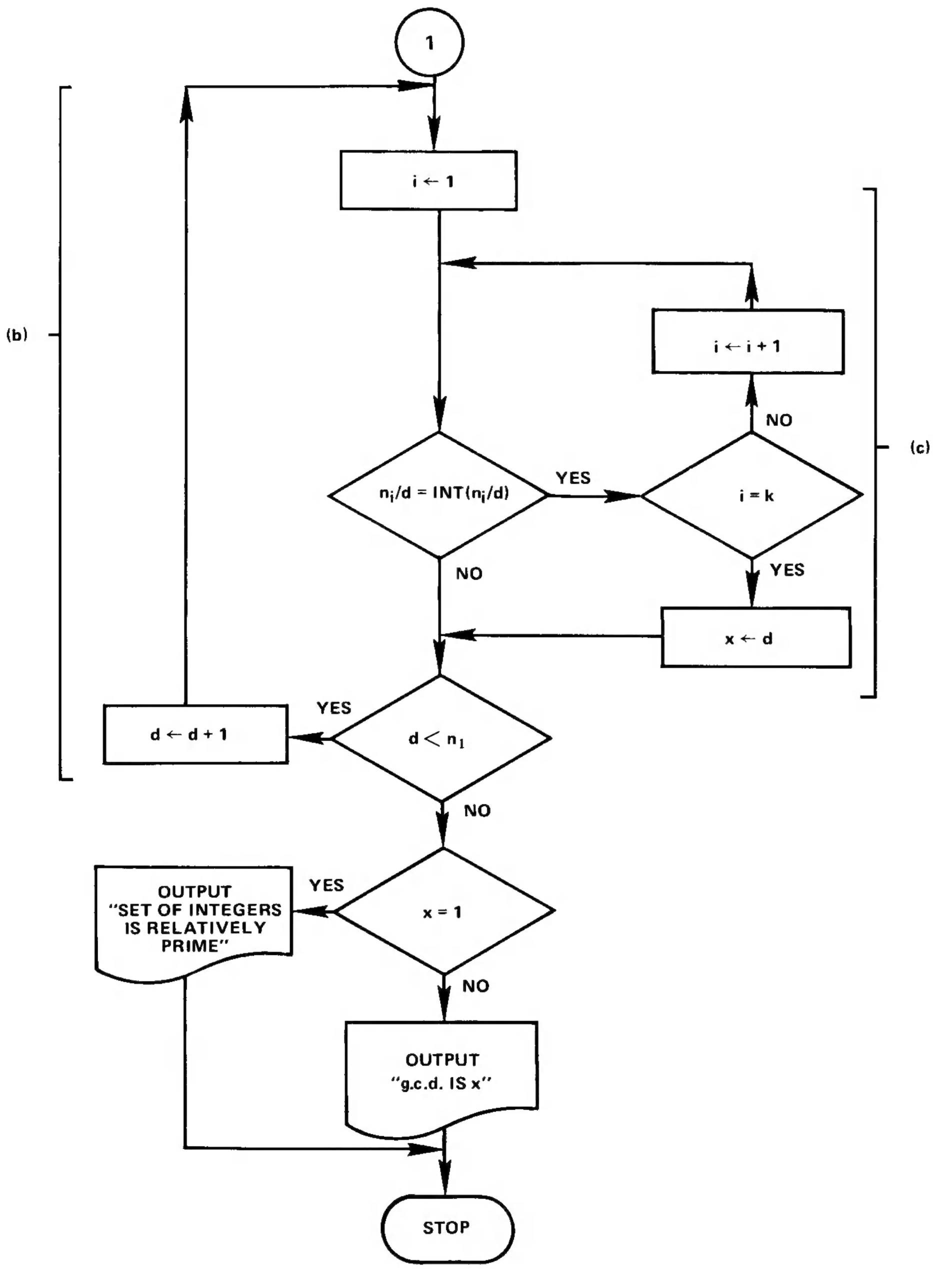
INTEGERS

EXERCISE 1 – Finding the Greatest Common Divisor

The loop indicated by (a) assigns subscripted variables to the integers in the set. The loop marked (b) selects the possible divisors of the set, the integers 1 through any number in the set. We pick the first number of the set as the upper limit of the possible divisors. If N_i/D is equal to the integer part of N_i/D, then we know that D is a divisor of N_i. The loop marked (c) determines if a possible divisor divides all integers in the set.

Micro Flow Chart




1
i ← 1
i ← i + 1
NO
(b)
(c)
n_i/d = INT(n_i/d)
YES
i = k
NO
YES
x ← d
YES
d ← d + 1
d < n_1
NO
OUTPUT
"SET OF INTEGERS
IS RELATIVELY
PRIME"
YES
x = 1
NO
OUTPUT
"g.c.d. IS x"
STOP

Example Program

Exercise 1.

```
10 REM--A PROGRAM TO DETERMINE IF INTERGERS OF A SET ARE RELATIVELY
20 REM--PRIME.
30 INPUT K
40 FOR I=1 TO K
50 READ N[I]
60 NEXT I
70 FOR D=1 TO N[1]
80 FOR I=1 TO K
90 IF N[I]/D#INT(N[I]/D) THEN 120
100 NEXT I
110 LET X=D
120 NEXT D
130 IF X=1 THEN 160
140 PRINT "G.C.D. IS";X
150 GOTO 180
160 PRINT "SET OF INTEGERS IS RELATIVELY PRIME."
170 DATA 91,26,169,286
180 END


RUN


?4
G.C.D.IS 13

DONE



170 DATA 71,251,149,353
RUN


?4
SET OF INTEGERS IS RELATIVELY PRIME.

DONE
```

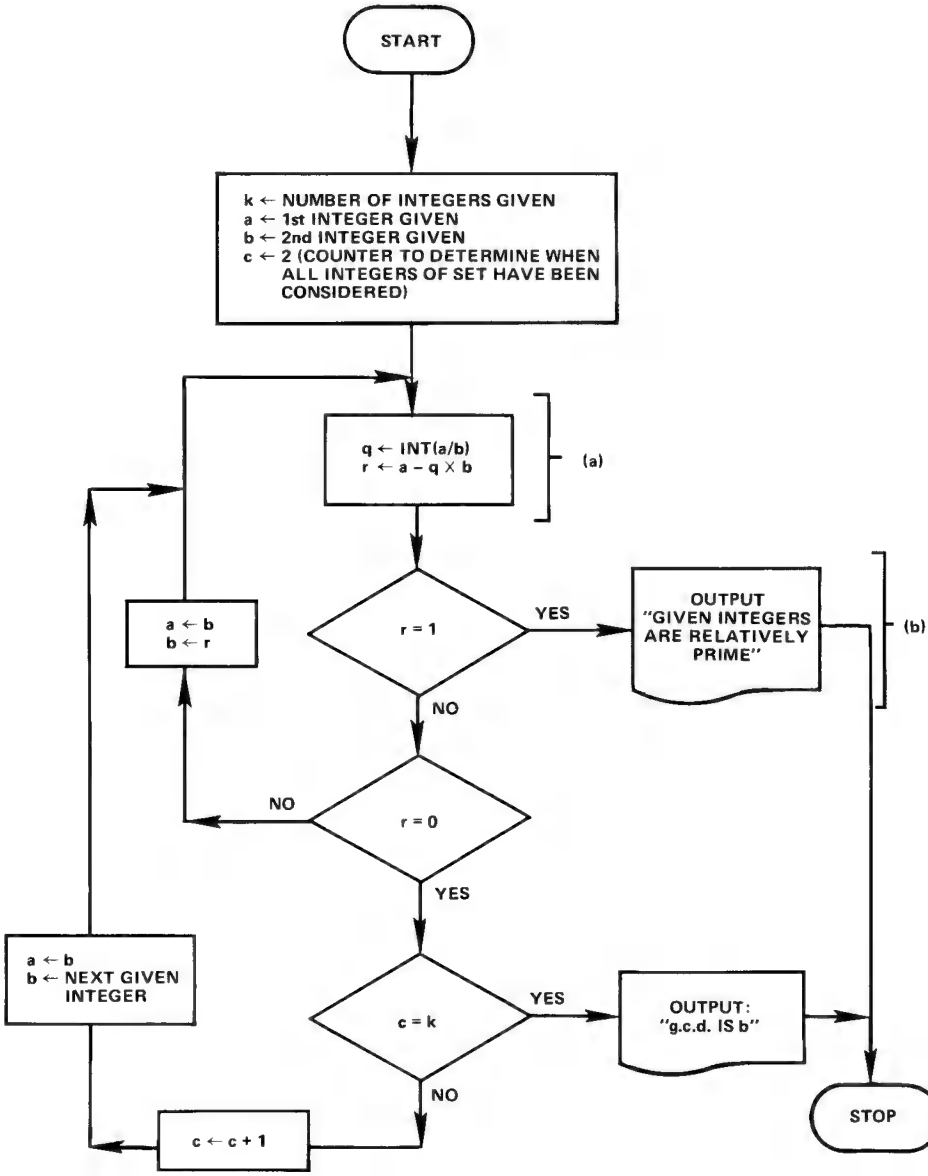
EXERCISE 2 – Using the Euclidean Algorithm

This problem is a good problem to demonstrate how the repetitive process of an algorithm is symbolized for programming purposes. The student should be able to write the program after careful study of the macro flow chart and the discussion in the Student Book. The student should recognize that the algorithm is applied in the following way:

(1) a ← first integer in given set, b ← second integer in given set
(2) Output g.c.d. of a and b
(3) a ← g.c.d. of a and b, b ← next integer in given set
(4) Go to (2)

The flow chart explains the procedure used. In the box labeled (a), the integral part of the quotient, q, for each a/b is determined and the remainder, r, is computed. Note that at (b) the entire set is labeled relatively prime if any two numbers being tested are found to be relatively prime.

Micro Flow Chart



Example Program

Exercise 2.

```
1   REM--A PROGRAM TO DETERMINE IF A SET OF ELEMENTS ARE RELATIVELY
2   REM--PRIME BY APPLYING THE EUCLIDEAN ALGORITHM.  ON THE DATA LINE
3   REM--ENTER THE NUMBER OF INTEGERS IN THE SET FOLLOWED BY THE INTEGERS
4   REM--IN THE SET
5   DATA 4,71,251,149,353
10   READ K,A,B
20   C=2
30   Q=INT(A/B)
40   R=A-Q*B
50   IF R=1 THEN 160
60   IF R=0 THEN 100
70   A=B
80   B=R
90   GOTO 30
100   IF C=K THEN 140
105   C=C+1
110   A=B
120   READ B
130   GOTO 30
140   PRINT "G.C.D.IS";B
150   GOTO 170
160   PRINT "GIVEN INTEGERS ARE RELATIVELY PRIME"
170   END
```

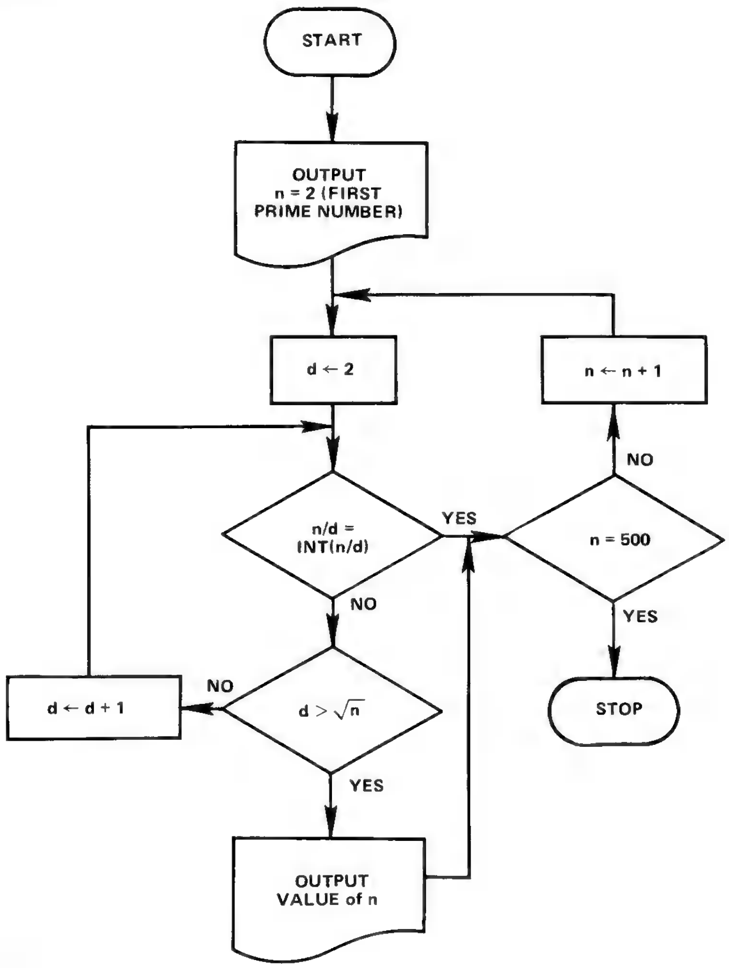
```
RUN


GIVEN INTEGERS ARE RELATIVELY PRIME

DONE
```

```
5 DATA 4,91,26,169,286
RUN


G.C.D.IS 13

DONE
```

EXERCISE 3 – Prime Numbers

Students should be able to complete this exercise without much assistance. Any questions will probably involve where to start their search and when to stop. We start with $n = 2$ because 2 is not considered a prime number. Our first divisor is 2, because every prime number has 1 as a divisor. We can stop when the divisor is greater than $\sqrt{n}$, since any divisor of n must also be a divisor of $\sqrt{n}$ ($n = a^2 = a \times a$; therefore any divisor of n is necessarily a divisor of a).

Micro Flow Chart



Example Program

Exercise 3.

```
5   REM--A PROGRAM TO FIND ALL PRIME NUMBERS FROM 1 TO 500.
10   N=2
15   GOTO 70
20   D=2
30   IF N/D=INT(N/D) THEN 80
40   IF D>SQR(N) THEN 70
50   D=D+1
60   GOTO 30
70   PRINT N;
80   IF N=500 THEN 110
90   N=N+1
100   GOTO 20
110   END
```

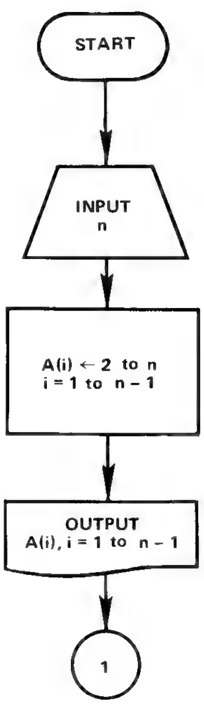
```
RUN

2     3     5     7     11    13    17    19    23    29    31    37
41    43    47    53    59    61    67    71    73    79    83    89
97    101   103   107   109   113   127   131   137   139   149   151
157   163   167   173   179   181   191   193   197   199   211   223
227   229   233   239   241   251   257   263   269   271   277   281
283   293   307   311   313   317   331   337   347   349   353   359
367   373   379   383   389   397   401   409   419   421   431   433
439   443   449   457   461   463   467   479   487   491   499
DONE
```

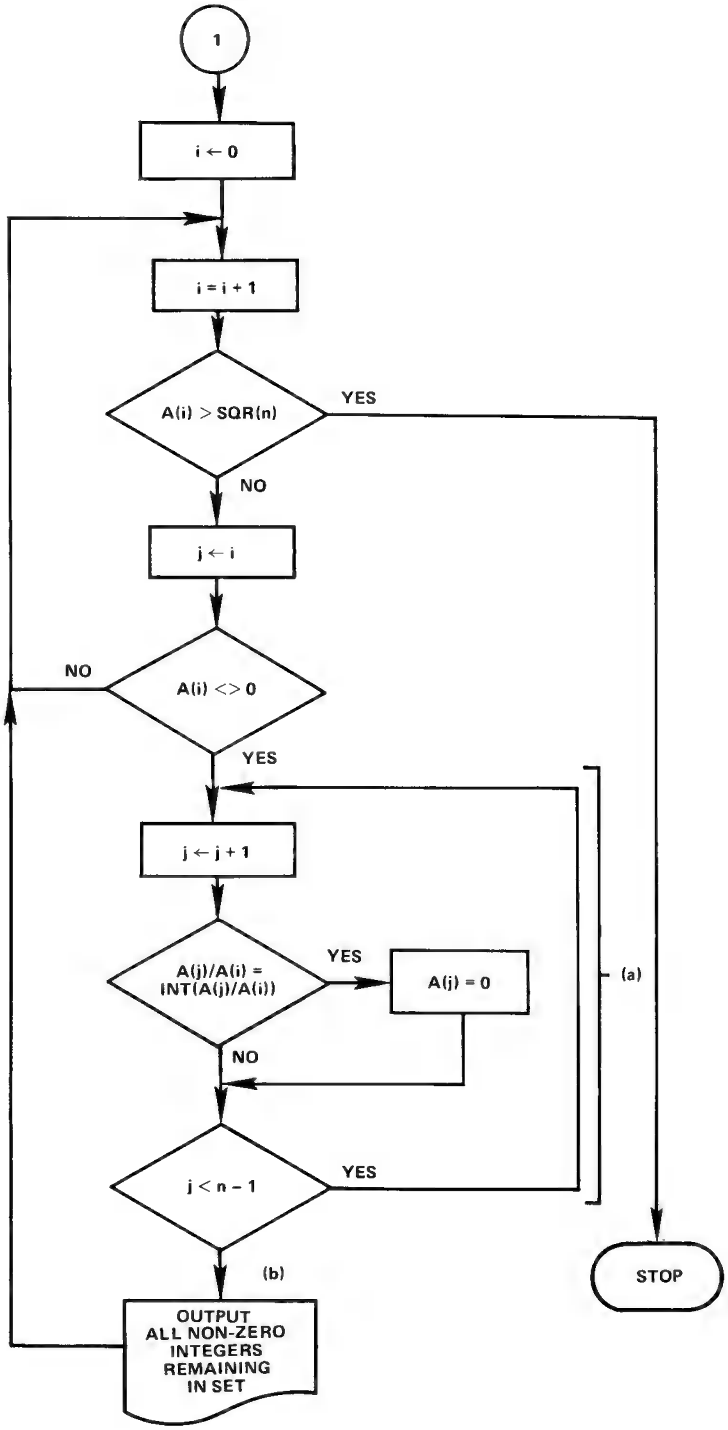
EXERCISE 4 – Determining Prime Numbers by "Sieve of Eratosthenes" Method

This problem requires the student to abstract an algorithm from the simulation of a given process. Make sure that your students actually use the "Sieve of Eratosthenes" method, i.e., eliminating multiples of each successive prime from 2 to $\sqrt{n}$, where n is the last integer in the set.

The flow chart below illustrates the correct application of this method. The loop marked (a) determines all multiples of a particular prime and sets them equal to zero. Each time the set is printed, elements equal to zero are omitted.

Micro Flow Chart




1
i ← 0
i = i + 1
A(i) > SQR(n)
YES
NO
j ← i
A(i) <> 0
NO
YES
j ← j + 1
A(j)/A(i) =
INT(A(j)/A(i))
YES
A(j) = 0
NO
j < n – 1
YES
(a)
(b)
OUTPUT
ALL NON-ZERO
INTEGERS
REMAINING
IN SET
STOP

Example Program

Exercise 4.

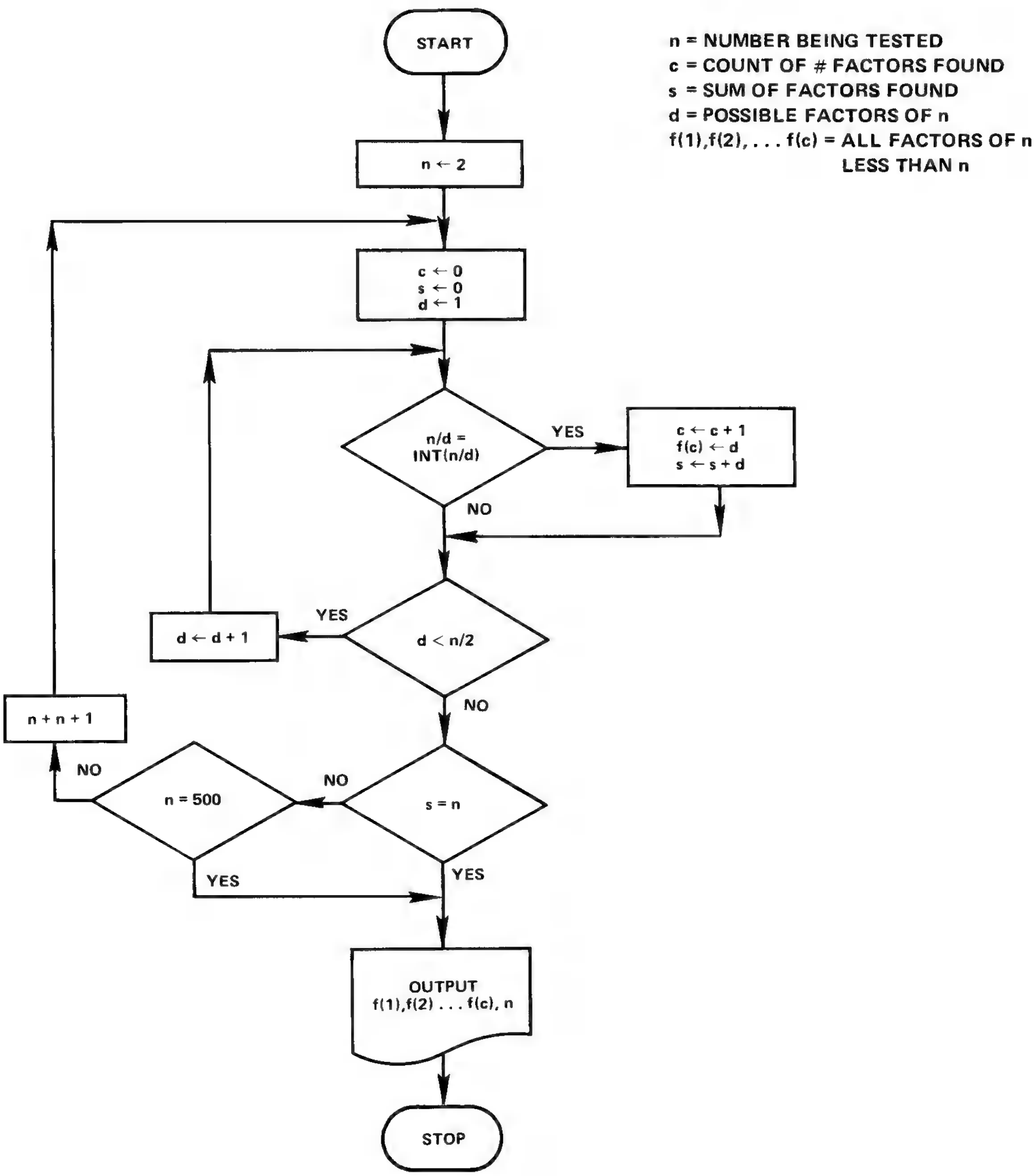
```
10  REM--A PROGRAM MODELING THE SIEVE OF ERATOSTHENES.  INPUT THE
20  REM--LARGEST NUMBER IN THE SIEVE.
30  INPUT N
40  DIM A[200]
50  FOR I=1 TO N-1
60  LET A[I]=I+1
70  PRINT A[I];
80  NEXT I
90  PRINT
100  PRINT
110  LET I=0
120  LET I=I+1
125  IF A[I]>SQR(N) THEN 280
130  LET J=I
140  IF A[I] <> 0 THEN 160
150  GOTO 120
160  J=J+1
170  IF A[J]/A[I]=INT(A[J]/A[I]) THEN 190
180  GOTO 200
190  A[J]=0
200  IF J<N-1 THEN 160
210  PRINT
220  PRINT
230  FOR K=1 TO N-1
240  IF A[K]=0 THEN 260
250  PRINT A[K];
260  NEXT K
270  GOTO 120
280  END
```

```
RUN


?30
 2     3     4     5     6     7     8     9     10    11    12    13
 14    15    16    17    18    19    20    21    22    23    24    25
 26    27    28    29    30


 2     3     5     7     9     11    13    15    17    19    21    23
 25    27    29

 2     3     5     7     11    13    17    19    23    25    29

 2     3     5     7     11    13    17    19    23    29
DONE
```

EXERCISE 5 – Determining Perfect Numbers

This problem introduces the student to interesting related concepts, i.e., Mersene Primes and Amicable Numbers. Hopefully, the student will be inspired to investigate some of the ideas he will come across when he consults the suggested references.

Micro Flow Chart



Example Program

Exercise 5.

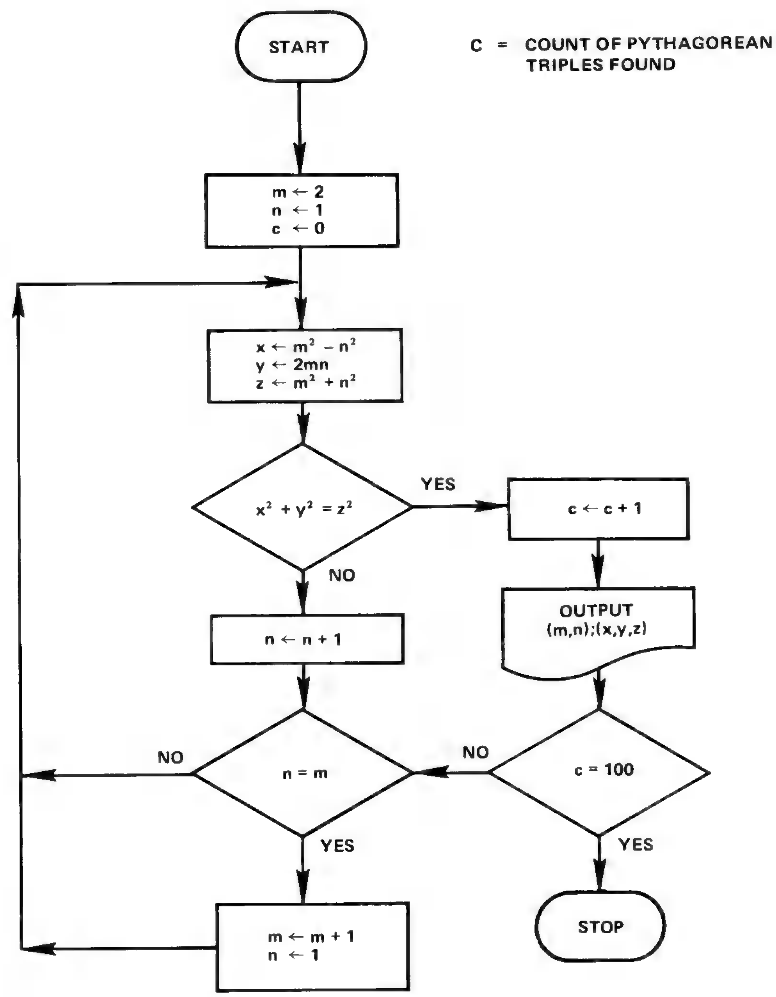
```
5  REM--A PROGRAM TO FIND ALL PERFECT NUMBERS LESS THAN OR EQUAL TO 500.
10  DIM F[50]
20  FOR N=2 TO 500
30  LET C=0
40  LET S=0
50  FOR D=1 TO N/2
60  IF N/D=INT(N/D) THEN 110
70  NEXT D
80  GOTO 150
90  NEXT N
100  GOTO 230
110  LET C=C+1
120  LET F[C]=D
130  LET S=S+D
140  GOTO 70
150  IF S=N THEN 170
160  GOTO 90
170  FOR I=1 TO C
180  PRINT F[I];",";
190  NEXT I
200  PRINT "ARE FACTORS OF THE PERFECT NUMBER"N
210  PRINT
220  GOTO 90
230  END
```

```
RUN

 1     , 2     , 3     ,ARE FACTORS OF THE PERFECT NUMBER 6

 1     , 2     , 4     , 7     , 14    ,ARE FACTORS OF THE PERFECT NUMBER 28

 1     , 2     , 4     , 8     , 16    , 31    , 62    , 124   , 248   ,
ARE FACTORS OF THE PERFECT NUMBER 496

DONE
```

EXERCISE 6 – Finding Pythagorean Triples

Have students research why Pythagorean triples (x,y,z) *must meet the relationships* $x = m^2 - n^2$, $y = 2mn$, *and* $z = m^2 + n^2$, m *and* n *positive integers,* $m > n$.

Micro Flow Chart



Example Program

Exercise 6(a).

```
5  REM--A PROGRAM TO FIND PYTHAGOREAN TRIPLES.
10  M=2
20  N=1
30  C=0
40  X=M↑2-N↑2
50  Y=2*M*N
60  Z=M↑2+N↑2
70  IF X↑2+Y↑2=Z↑2 THEN 140
80  LET N=N+1
90  IF N=M THEN 110
100  GOTO 40
110  LET M=M+1
120  LET N=1
125  IF N=M THEN 90
130  GOTO 40
140  C=C+1
150  PRINT "(";M;",";N;")""(";X;",";Y;",";Z;")"
160  IF C=100 THEN 180
170  GOTO 80
180  END
```

```
RUN

( 2     , 1     )( 3     , 4     , 5     )
( 3     , 1     )( 8     , 6     , 10    )
( 3     , 2     )( 5     , 12    , 13    )
( 4     , 1     )( 15    , 8     , 17    )
( 4     , 2     )( 12    , 16    , 20    )
( 4     , 3     )( 7     , 24    , 25    )
( 5     , 1     )( 24    , 10    , 26    )
( 5     , 2     )( 21    , 20    , 29    )
( 5     , 3     )( 16    , 30    , 34    )
( 5     , 4     )( 9     , 40    , 41    )
( 6     , 1     )( 35    , 12    , 37    )
( 6     , 2     )( 32    , 24    , 40    )
( 6     , 3     )( 27    , 36    , 45    )
( 6     , 4     )( 20    , 48    , 52    )
( 6     , 5     )( 11    , 60    , 61    )
( 7     , 1     )( 48    , 14    , 50    )
( 7     , 2     )( 45    , 28    , 53    )
( 7     , 3     )( 40    , 42    , 58    )
( 7     , 4     )( 33    , 56    , 65    )
( 7     , 5     )( 24    , 70    , 74    )
( 7     , 6     )( 13    , 84    , 85    )
( 8     , 1     )( 63    , 16    , 65    )
( 8     , 2     )( 60    , 32    , 68    )
( 8     , 3     )( 55    , 48    , 73    )
( 8     , 4     )( 48    , 64    , 80    )
( 8     , 5     )( 39    , 80    , 89    )
( 8     , 6     )( 28    , 96    , 100   )
( 8     , 7     )( 15    , 112   , 113   )
( 9     , 1     )( 80    , 18    , 82    )
( 9     , 2     )( 77    , 36    , 85    )
( 9     , 3     )( 72    , 54    , 90    )
( 9     , 4     )( 65    , 72    , 97    )
( 9     , 5     )( 56    , 90    ,
STOP
```

Example Program

Exercise 6(b).

```
10   REM--A PROGRAM TO FIND PRIMITIVE PYTHAGOREAN TRIPLES
20   M=2
30   N=1
40   C=0
50   X=M↑2-N↑2
60   Y=2*M*N
70   Z=M↑2+N↑2
80   IF X↑2+Y↑2=Z↑2 THEN 160
90   LET N=N+1
100   IF N=M THEN 120
110   GOTO 50
120   LET M=M+1
130   LET N=1
140   GOTO 50
160   LET A=X
170   LET B=Y
180   Q=INT(A/B)
190   R=A-Q*B
200   IF R=1 THEN 250
210   IF R=0 THEN 90
220   A=B
230   B=R
240   GOTO 180
250   C=C+1
260   PRINT "(";M;",";N;")""(";X;",";Y;",";Z;")"
270   IF C=100 THEN 290
280   GOTO 90
290   END


RUN

( 2     , 1     )( 3     , 4     , 5     )
( 3     , 2     )( 5     , 12    , 13    )
( 4     , 1     )( 15    , 8     , 17    )
( 4     , 3     )( 7     , 24    , 25    )
( 5     , 2     )( 21    , 20    , 29    )
( 5     , 4     )( 9     , 40    , 41    )
( 6     , 1     )( 35    , 12    , 37    )
( 6     , 5     )( 11    , 60    , 61    )
( 7     , 2     )( 45    , 28    , 53    )
( 7     , 4     )( 33    , 56    , 65    )
( 7     , 6     )( 13    , 84    , 85    )
( 8     , 1     )( 63    , 16    , 65    )
( 8     , 3     )( 55    , 48    , 73    )
( 8     , 5     )( 39    , 80    , 89    )
( 8     , 7     )( 15    , 112   , 113   )
( 9     , 2
STOP
```

RATIONAL NUMBERS

EXERCISE 7 – Denumeration of the Set of Positive Rational Numbers

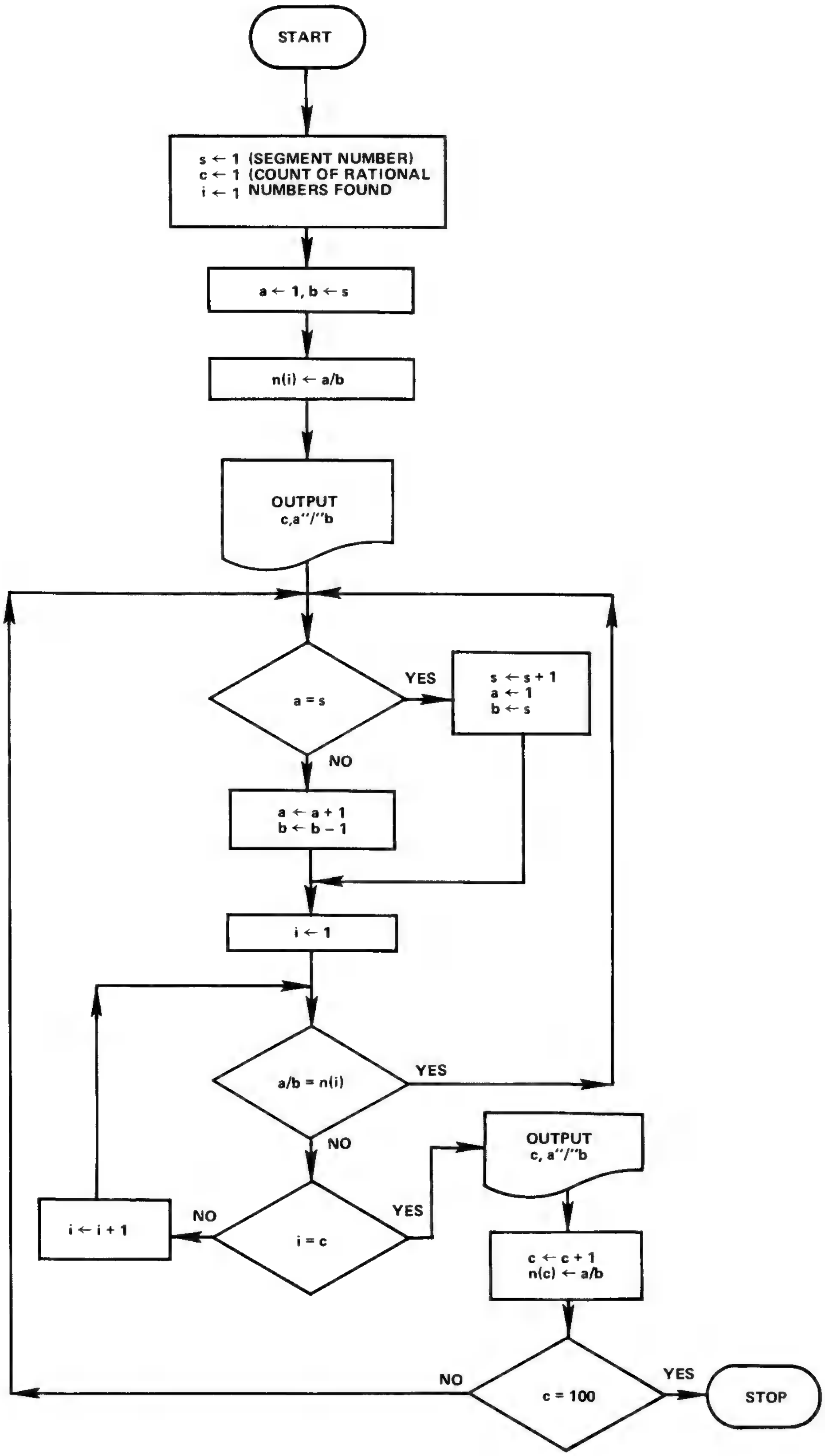
This exercise is another which requires that the student formulate an algorithm from the output pattern generated by hand.

By reordering the terms in alternating segments (diagonals) of the sequence, we have a new sequence in which the terms in every segment are in increasing order:

$$1/1 \mid 1/2,\ 2/1 \mid 1/3,\ 2/2,\ 3/1 \mid 1/4,\ 2/3,\ 3/2,\ 4/1 \mid$$
$$\mid 1/5,\ 2/4,\ 3/3,\ 4/2,\ 5/1 \mid 1/6,\ 2/5,\ 3/4, \ldots$$

If you look closer at this sequence, you see that the numerators within each segment are increasing, while the denominators are decreasing. In general, then, we see that in segment s, the numbers are $1/s$, $2/(s-1)$, $3/(s-2) \ldots (s-2)/3$, $(s-1)/2$, $s/1$.

In the flow chart, note that a"/"b represents the value of a "over" the value of b, and a/b represents the quotient of a divided by b.


START
s ← 1 (SEGMENT NUMBER)
c ← 1 (COUNT OF RATIONAL
i ← 1 NUMBERS FOUND
a ← 1, b ← s
n(i) ← a/b
OUTPUT
c,a"/"b
a = s
YES
s ← s + 1
a ← 1
b ← s
NO
a ← a + 1
b ← b − 1
i ← 1
a/b = n(i)
YES
NO
OUTPUT
c, a"/"b
NO
i ← i + 1
i = c
YES
c ← c + 1
n(c) ← a/b
NO
c = 100
YES
STOP

Example Program

Exercise 7(a).

```
1   DIM N[100]
5   REM--A PROGRAM TO DEMONSTRATE THE DENUMERABILITY OF THE SET
6   REM-- RATIONAL NUMBERS.
10  S=1
20  I=1
30  C=1
40  A=1
50  B=S
55  N[I]=A/B
60  PRINT C;A"/"B
70  IF A=S THEN 190
80  A=A+1
90  B=B-1
100  I=1
110  IF A/B=N[I] THEN 70
120  IF I=C THEN 150
130  I=I+1
140  GOTO 110
150  C=C+1
160  PRINT C;A"/"B
165  LET N[C]=A/B
170  IF C=100 THEN 230
180  GOTO 70
190  S=S+1
200  A=1
210  B=S
220  GOTO 100
230  END
```

```
RUN

 1       1       / 1
 2       1       / 2
 3       2       / 1
 4       1       / 3
 5       3       / 1
 6       1       / 4
 7       2       / 3
 8       3       / 2
 9       4       / 1
 10      1       / 5
 11      5       / 1
 12      1       / 6
 13      2       / 5
 14      3       / 4
 15      4       / 3
 16      5       / 2
 17      6       / 1
 18      1       / 7
 19      3       / 5
 20      5       / 3
 21      7       / 1
 22      1       / 8
 23      2       / 7
 24      4       / 5
 25      5       / 4
 26      7       / 2
 27      8       / 1
 28      1       / 9
 29      3       / 7
 30      7       / 3
 31      9       / 1
 32      1       / 10
 33      2       / 9
 34      3       / 8

STOP
```

Exercise 7(b).

```
10  DIM N[100]
20  I=1
30  REM--A PROGRAM TO DEMONSTRATE THE DENUMERABILITY OF THE SET
40  REM-- RATIONAL NUMBERS.
50  PRINT 1;0"/"1
60  N[I]=0/1
70  S=1
80  I=1
90  C=2
100  A=1
110  B=S
120  N[C]=A/B
130  PRINT C;A"/"B
140  I=I+1
150  PRINT C;"-"A"/"B
160  IF A=S THEN 320
170  A=A+1
180  B=B-1
190  I=1
200  IF A/B=N[I] THEN 160
210  IF I=C THEN 240
220  I=I+1
230  GOTO 200
240  C=C+1
250  PRINT C;A"/"B
260  LET N[C]=A/B
270  C=C+1
280  PRINT C;"-"A"/"B
290  N[C]=A/B
300  IF C=100 THEN 360
310  GOTO 160
320  S=S+1
330  A=1
340  B=S
350  GOTO 190
360  END
```

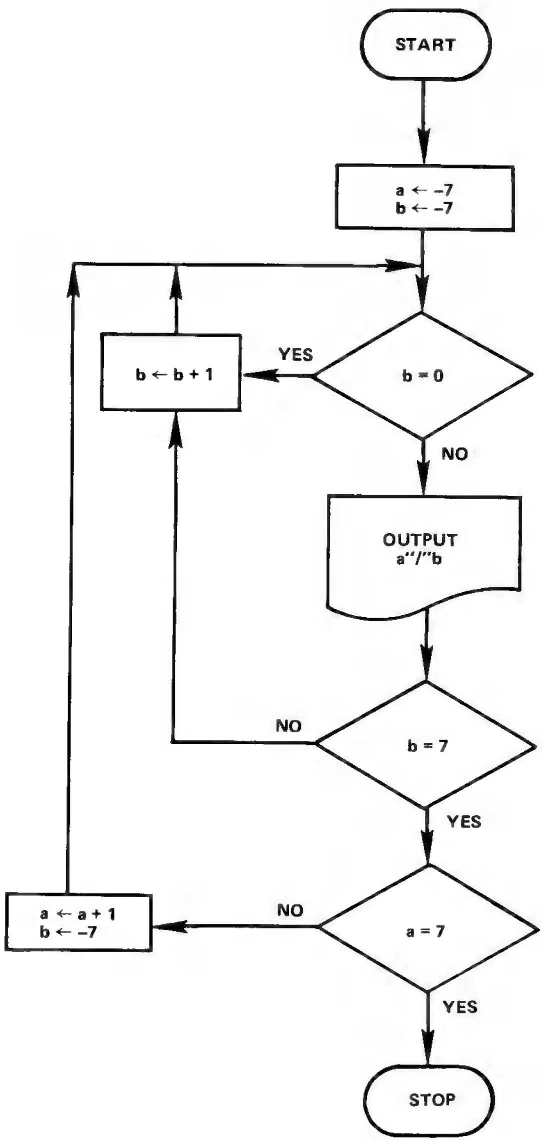
```
RUN

 1     0    / 1
 2     1    / 1
 2    - 1    / 1
 3     1    / 2
 4    - 1    / 2
 5     2    / 1
 6    - 2    / 1
 7     1    / 3
 8    - 1    / 3
 9     3    / 1
 10   - 3    / 1
 11    1    / 4
 12   - 1    / 4
 13    2    / 3
 14   - 2    / 3
 15    3    / 2
 16   - 3    / 2
 17    4    / 1
 18   - 4    / 1
STOP
```

EXERCISE 8 – Forming Rational Numbers, Using All Integers in a Given Real Number Interval

This program is significant because it models the definition of a rational number. The student will have to deal with the condition that division by zero is not possible.

Micro Flow Chart

Part (a)



Example Program

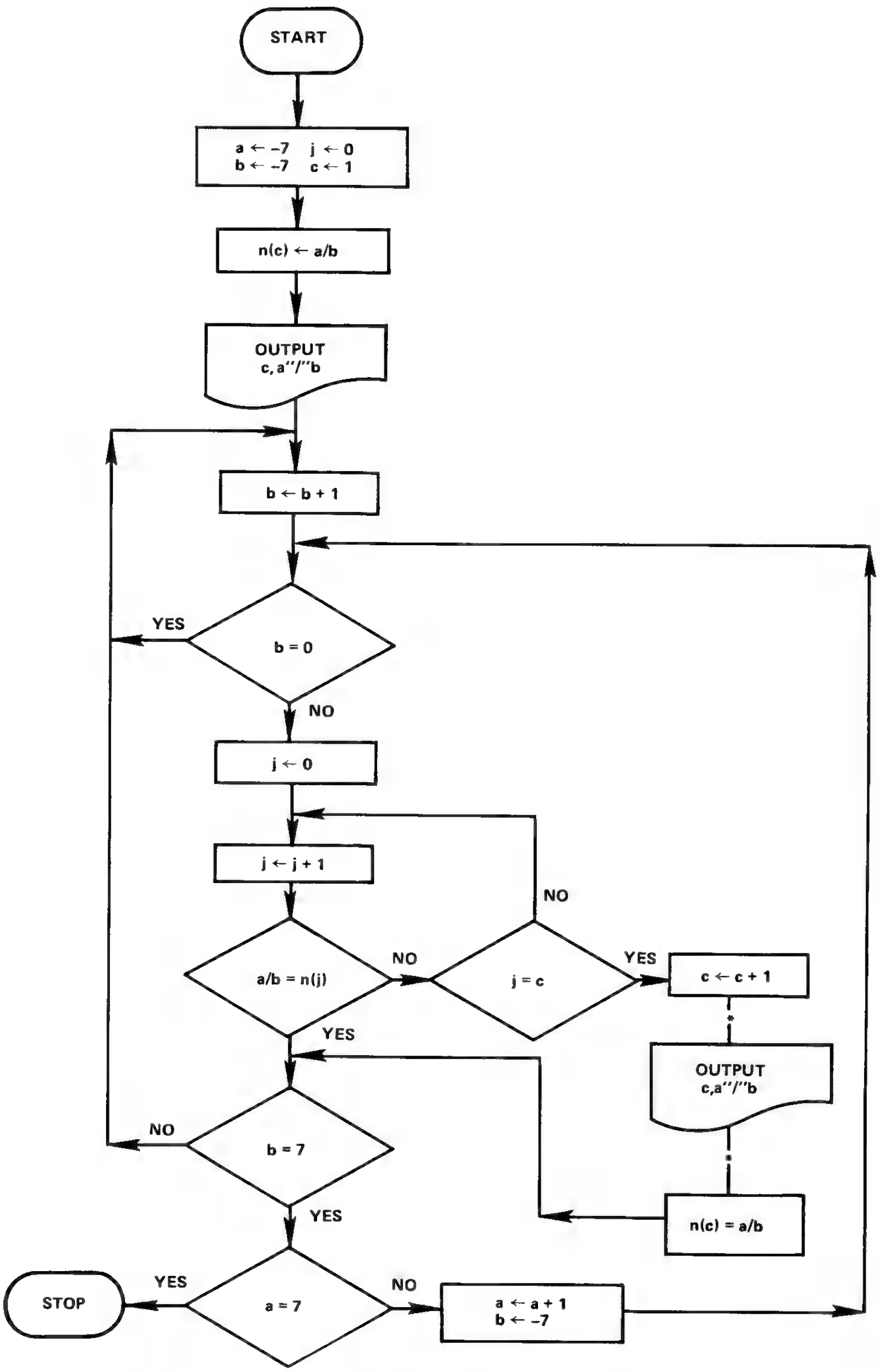
Exercise 8(a).

```
5   REM--A PROGRAM TO FORM ALL POSSIBLE RATIONAL NUMBERS
6   REM--USING THE INTEGERS OVER A GIVEN REAL NUMBER INTERVAL.
10   FOR A=-7 TO 7
20   FOR B=-7 TO 7
30   IF B=0 THEN 50
40   PRINT A"/"B,
50   NEXT B
60   NEXT A
80   END
```

```
RUN

-7    /-7     -7    /-6     -7    /-5     -7    /-4     -7    /-3
-7    /-2     -7    /-1     -7    / 1     -7    / 2     -7    / 3
-7    / 4     -7    / 5     -7    / 6     -7    / 7     -6    /-7
-6    /-6     -6    /-5     -6    /-4     -6    /-3     -6    /-2
-6    /-1     -6    / 1     -6    / 2     -6    / 3     -6    / 4
-6    / 5     -6    / 6     -6    / 7     -5    /-7     -5    /-6
-5    /-5     -5    /-4     -5    /-3     -5    /-2     -5    /-1
-5    / 1     -5    / 2     -5    / 3     -5    / 4     -5    / 5
-5    / 6     -5    / 7     -4    /-7     -4    /-6     -4    /-5
-4    /-4     -4    /-3     -4    /-2     -4    /-1     -4    / 1
-4    / 2     -4    / 3     -4    / 4     -4    / 5     -4    / 6
-4    / 7     -3    /-7     -3    /-6     -3    /-5     -3    /-4
-3    /-3     -3    /-2     -3    /-1     -3    / 1     -3    / 2
-3    / 3     -3    / 4     -3    / 5     -3    / 6     -3    / 7
-2    /-7     -2    /-6     -2    /-5     -2    /-4     -2    /-3
-2    /-2     -2    /-1     -2    / 1     -2    / 2     -2    / 3
-2    / 4     -2    / 5     -2    / 6     -2    / 7     -1    /-7
-1    /-6     -1    /-5     -1    /-4     -1    /-3     -1    /-2
-1    /-1     -1    / 1     -1    / 2     -1    / 3     -1    / 4
-1    / 5     -1    / 6     -1    / 7      0    /-7      0    /-6
 0    /-5      0    /-4      0    /-3      0    /-2      0    /-1
 0    / 1      0    / 2      0    / 3      0    / 4      0    / 5
 0    / 6      0    / 7      1    /-7      1    /-6      1    /-5
 1    /-4      1    /-3      1    /-2      1    /-1      1    / 1
 1    / 2      1    / 3      1    / 4      1    / 5      1    / 6
```

Micro Flow Chart

Part (b).



(*) For Part (c). A section of flow chart will replace this flow chart between the *.

Example Program

Exercise 8(b)

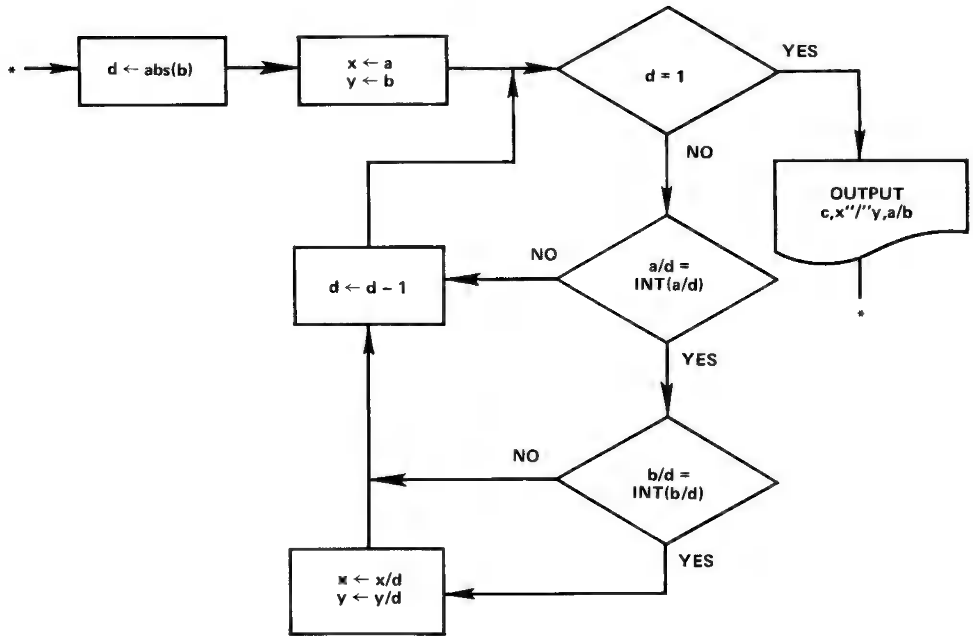
```
5   REM-- A PROGRAM THAT WILL PRINT ALL MEANINGFUL RATIONAL
6   REM-- EXPRESSIONS THAT CAN BE FORMED FROM THE INTEGERS OVER
7   REM--A REAL NUMBER INTERVAL. EXPRESSIONS THAT HAVE THE SAME
9   REM--VALUE AS SOME EXPRESSION ALREADY PRINTED WILL NOT BE PRINTED.
10  READ A,B,C,J
15  DIM N[100]
20  DATA -7,-7,1,0
40  PRINT C;A"/"B
46  LET N[C]=A/B
50  LET B=B+1
60  IF B=0 THEN 50
65  J=0
70  LET J=J+1
80  IF A/B=N[J] THEN 110
90  IF J=C THEN 125
100  GOTO 70
110  IF B=7 THEN 170
120  GOTO 50
125  C=C+1
130  PRINT C;A"/"B
150  LET N[C]=A/B
160  GOTO 110
170  IF A=7 THEN 210
180  LET A=A+1
190  LET B=-7
200  GOTO 60
210  END
```

```
RUN

 1    -7    /-7
 2    -7    /-6
 3    -7    /-5
 4    -7    /-4
 5    -7    /-3
 6    -7    /-2
 7    -7    /-1
 8    -7    / 1
 9    -7    / 2
 10   -7    / 3
 11   -7    / 4
 12   -7    / 5
 13   -7    / 6
 14   -7    / 7
 15   -6    /-7
 16   -6    /-5
 17   -6    /-4
 18   -6    /-3
 19   -6    /-2
STOP
```

Micro Flow Chart

Part (c).

Insert the following flow chart section as indicated in flow chart of Part (b).



Example Program

Exercise 8(c).

```
10  REM--A PROGRAM TO PRINT OUT IN REDUCED FORM AND DECIMAL FORM
20  REM--ALL RATIONAL NUMBERS THAT CAN BE FORMED FROM THE INTEGERS
30  REM --OVER SOME REAL NUMBER INTERVAL.
40  READ A,B,C,J
50  DIM N[100]
60  DATA -7,-7,1,0
70  PRINT C;A"/"B;A/B
80  LET N[C]=A/B
90  LET B=B+1
100  IF B=0 THEN 90
110  J=0
120  LET J=J+1
130  IF A/B=N[J] THEN 160
140  IF J=C THEN 180
150  GOTO 120
160  IF B=7 THEN 310
170  GOTO 90
180  C=C+1
190  LET D=ABS(B)
191  LET X=A
192  LET Y=B
195  IF D=1 THEN 280
200  IF A/D=INT(A/D) THEN 240
220  LET D=D-1
230  GOTO 195
240  IF B/D=INT(B/D) THEN 260
250  GOTO 220
260  LET X=X/D
270  LET Y=Y/D
275  GOTO 220
280  PRINT C;X"/"Y;A/B
290  LET N[C]=A/B
300  GOTO 160
310  IF A=7 THEN 350
320  LET A=A+1
330  LET B=-7
340  GOTO 100
350  END
```

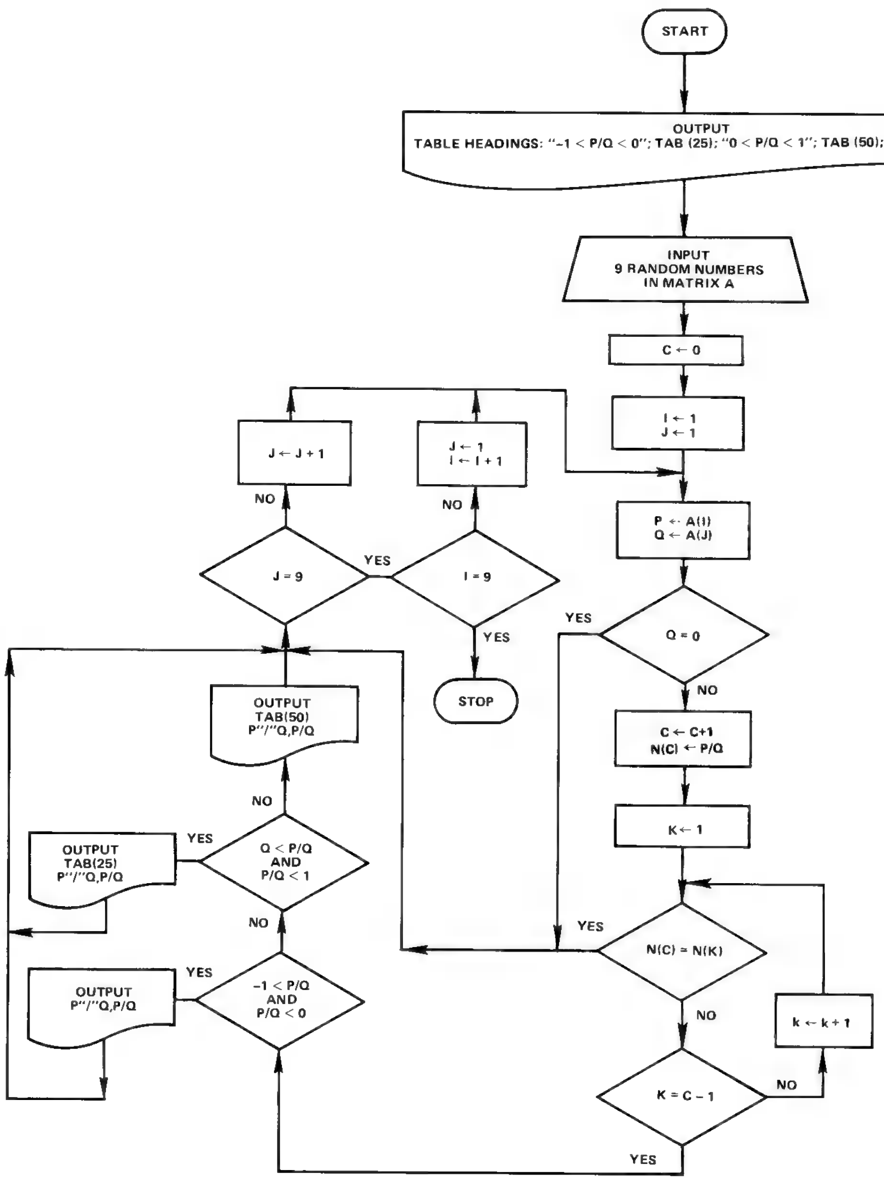
RUN

```
 1    -7    /-7     1
 2    -7    /-6     1.16667
 3    -7    /-5     1.4
 4    -7    /-4     1.75
 5    -7    /-3     2.33333
 6    -7    /-2     3.5
 7    -7    /-1     7
 8    -7    / 1    -7
 9    -7    / 2    -3.5
 10   -7    / 3    -2.33333
 11   -7    / 4    -1.75
 12   -7    / 5    -1.4
 13   -7    / 6    -1.16667
 14   -1    / 1    -1
 15   -6    /-7     .857143
 16   -6    /-5     1.2
 17   -3    /-2     1.5
 18   -2    /-1     2
 19   -3    /-1     3
 20   -6    /-1     6
```

EXERCISE 9 – Rational Numbers Formed from a Random Set of Integers

This problem is a more complicated case of Exercise 8. It illustrates that the magnitude of the rational number formed is determined by the relationship of numerator to denominator and the sign of each. Also, the idea of a decimal approximation to a rational number is introduced.

Micro Flow Chart



NOTE: TAB (X) INDICATES MOVING CARRIAGE TO Xth POSITION ON THE PAGE.

Example Program

Exercise 9.

```
10  REM-- A PROGRAM THAT SEPARATES RATIONAL NUMBERS FORMED INTO
20  REM  THREE DIFFERENT CATAGORIES.
30  PRINT "-1<P/Q<0";TAB(25);"0<P/Q<1";TAB(50);"ABS(P/Q)>1"
40  DIM N[100]
50  FOR I=1 TO 9
60  READ A[I]
70  NEXT I
80  DATA -16,3,45,0,-24,-6,7,-19,38
90  LET C=0
100  FOR I=1 TO 9
110  FOR J=1 TO 9
120  LET P=A[I]
130  LET Q=A[J]
140  IF Q=0 THEN 270
150  LET C=C+1
160  LET N[C]=P/Q
170  FOR K=1 TO C-1
180  IF N[C]=N[K] THEN 270
190  NEXT K
200  IF -1<P/Q AND P/Q<0 THEN 240
210  IF 0<P/Q AND P/Q<1 THEN 260
220  PRINT TAB(50);P;"/";Q;P/Q
230  GOTO 270
240  PRINT P;"/";Q;P/Q
250  GOTO 270
260  PRINT TAB(25);P;"/";Q;P/Q
270  NEXT J
280  NEXT I
290  END
```

RUN

```
-1<P/Q<0                     0<P/Q<1                        ABS(P/Q)>1
                                                            -16    /-16     1
                                                            -16    / 3     -5.33333
-16    / 45    -.355556
                             -16    /-24    .666667
                                                            -16    /-6      2.66667
                                                            -16    / 7     -2.28571
                             -16    /-19    .842105
-16    / 38    -.421053
 3     /-16    -.1875
                              3     / 45    6.66667E-02
 3     /-24    -.125
 3     /-6     -.5
                              3     / 7     .428571
 3     /-19    -.157895
                              3     / 38    7.89474E-02
                                                             45    /-16    -2.8125
                                                             45    / 3      15
                                                             45    /-24    -1.875
                                                             45    /-6     -7.5
                                                             45    / 7      6.42857
                                                             45    /-19    -2.36842
                                                             45    / 38     1.18421
                                                             0     /-16     0
                                                            -24    /-16     1.5
                                                            -24    / 3     -8
-24    / 45    -.533333
                                                            -24    /-6      4
                                                            -24    / 7     -3.42857
                                                            -24    /-19     1.26316
-24    / 38    -.631579
                             -6     /-16    .375
                                                            -6     / 3     -2
-6     / 45    -.133333
                             -6     /-24    .25
-6     / 7     -.857143
                             -6     /-19    .315789
 7     /-16    -.4375
                                                             7     / 3      2.33333
                              7     / 45    .155556
 7     /-24    -.291667
                                                             7     /-6     -1.16667
 7     /-19    -.368421
                              7     / 38    .184211
                                                            -19    /-16     1.1875
                                                            -19    / 3     -6.33333
-19    / 45    -.422222
                             -19    /-24    .791667
                                                            -19    /-6      3.16667
                                                            -19    / 7     -2.71429
                                                             38    /-16    -2.375
                                                             38    / 3      12.6667
                              38    / 45    .844444
                                                             38    /-24    -1.58333
                                                             38    / 7      5.42857
```

DONE

IRRATIONAL NUMBERS

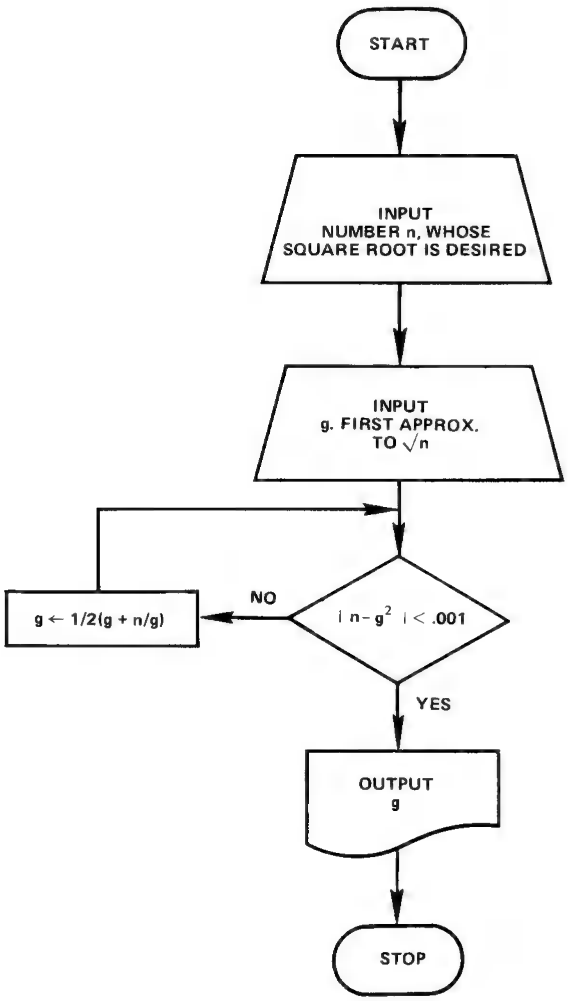
EXERCISE 10 – A Rational Approximation of $\sqrt{n}$, n a rational number > 0

Newton's method (averaging) of finding the square root of a number has become the most popular square root algorithm used in algebra texts. Your students will appreciate the opportunity to use the computer to perform the tedious and cumbersome arithmetic operations involved.

In Parts (b), (c), and (d), it is important that your students understand the development of the equations defining the algorithms used in each program. The exercises themselves are subordinate to this understanding. Part (d) will be most valuable if the students are expected to research the algorithm given for the solution.

Micro Flow Chart

Part (a).



Example Program

Exercise 10(a).

```
10  REM--A PROGRAM TO FIND THE SQUARE ROORT OF A NUMBER BY
15  REM--NEWTON'S METHOD
20  INPUT N,G
30  IF ABS(N-G↑2)<.001 THEN 60
40  LET G=(G+N/G)/2
45  PRINT G
50  GOTO 30
60  PRINT "APPROXIMATE SQUARE ROOT OF";N;"IS";G
70  END


RUN


?5,3
 2.33333
 2.2381
 2.23607
APPROXIMATE SQUARE ROOT OF 5    IS 2.23607

DONE
RUN


?16,5
 4.1
 4.00122
 4
APPROXIMATE SQUARE ROOT OF 16   IS 4

DONE
RUN


?16,3
 4.16667
 4.00333
 4.
APPROXIMATE SQUARE ROOT OF 16   IS 4.

DONE
RUN


?72,5
 9.7
 8.56134
 8.48562
 8.48528
APPROXIMATE SQUARE ROOT OF 72   IS 8.48528

DONE
```

Example Program

Exercise 10(b).

```
10  REM--APROGRAM TO FIND THE CUBE ROOT OF A NUMBER USING
20  REM--NEWTON'S METHOD.
30  INPUT N,G
40  IF ABS(N-G↑3)<.001 THEN 70
50  LET G=(G+N/G↑2)/2
55  PRINT G
60  GOTO 40
70  PRINT "APPROXIMATE CUBE ROOT OF";N;"IS";G
80  END


RUN


?27,5
 3.04
 2.98079
 3.00979
 2.99515
 3.00244
 2.99879
 3.00061
 2.9997
 3.00015
 2.99992
 3.00004
 2.99998
APPROXIMATE CUBE ROOT OF 27   IS 2.99998

DONE
RUN


?27,3
APPROXIMATE CUBE ROOT OF 27   IS 3

DONE
RUN


?-8,4
 1.75
-.431123
-21.7364
-10.8766
-5.47214
-2.86965
-1.92056
-2.04471
-1.9791
-2.01078
-1.99469
-2.00267
-1.99867
-2.00067
-1.99967
-2.00017
-1.99992
APPROXIMATE CUBE ROOT OF-8    IS-1.99992

DONE
```

Example Program

Exercise 10(d).

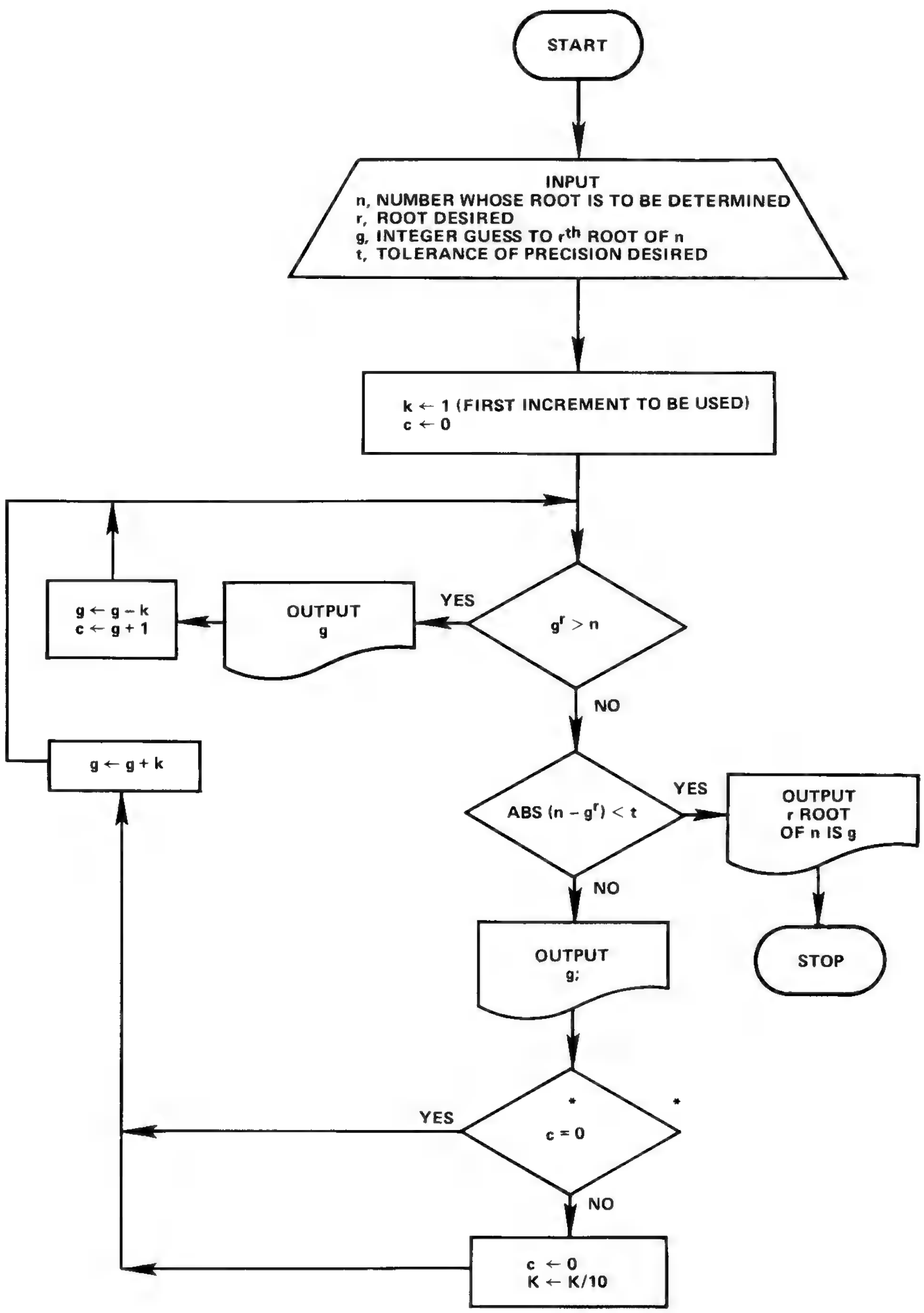
```
5  REM--A PROGRAM TO COMPUTE THE CUBE ROOT OF A NUMBER.
10  INPUT N,G
20  LET H=1/3*(2*G+N/G↑2)
25  PRINT H;
30  IF ABS(N-H↑3)<.001 THEN 60
40  LET G=H
50  GOTO 20
60  PRINT "THE CUBE ROOT OF ";N;"IS";H
70  END


RUN


?16,2
 2.66667     2.52778     2.51987    THE CUBE ROOT OF  16   IS 2.51987

DONE
RUN


?27,5
 3.69333     3.12201     3.00471     3.00001    THE CUBE ROOT OF  27
IS 3.00001

DONE
RUN


?63,7
 5.09524     4.20572     3.99105     3.97909     3.97906
THE CUBE ROOT OF  63   IS 3.97906

DONE
RUN


?-8,4
 2.5         1.24        -.907638    -3.8421     -2.74205    -2.1827
-2.01487    -2.00011     -2     THE CUBE ROOT OF -8    IS-2

DONE
```

*EXERCISE 11 – **Incrementing Method for Finding** $\sqrt{n}$, **n Rational***

This problem is valuable in that it can serve as an introduction to the concept of a limit and the various root search methods used to find roots of polynomial equations.

The variable c is kept and incremented if $g > \sqrt[3]{n}$. *Later it is tested for 0 to determine whether the increment should be changed.*

Micro Flow Chart



Example Program

Exercise 11.

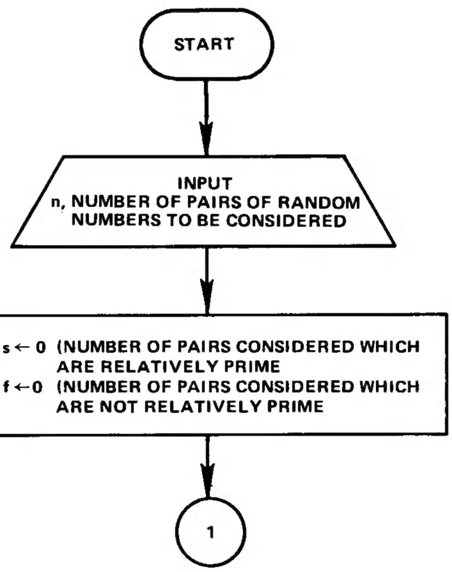
```
5   REM--A PROGRAM FOR FINDING THE RTH. ROOT OF A NUMBER BY AN
6   REM--ITERATIVE PROCESS.
10   INPUT N,R,G,T
20   LET K=1
30   LET C=0
40   IF G↑R>N THEN 120
50   IF ABS(N-G↑R)<T THEN 160
60   PRINT G;
70   IF C=0 THEN 100
80   LET C=0
90   LET K=K/10
100   LET G=G+K
110   GOTO 40
120   PRINT G;
130   LET G=G-K
140   LET C=C+1
150   GOTO 40
160   PRINT R;"TH. ROOT OF ";N;"IS";G
170   END
```

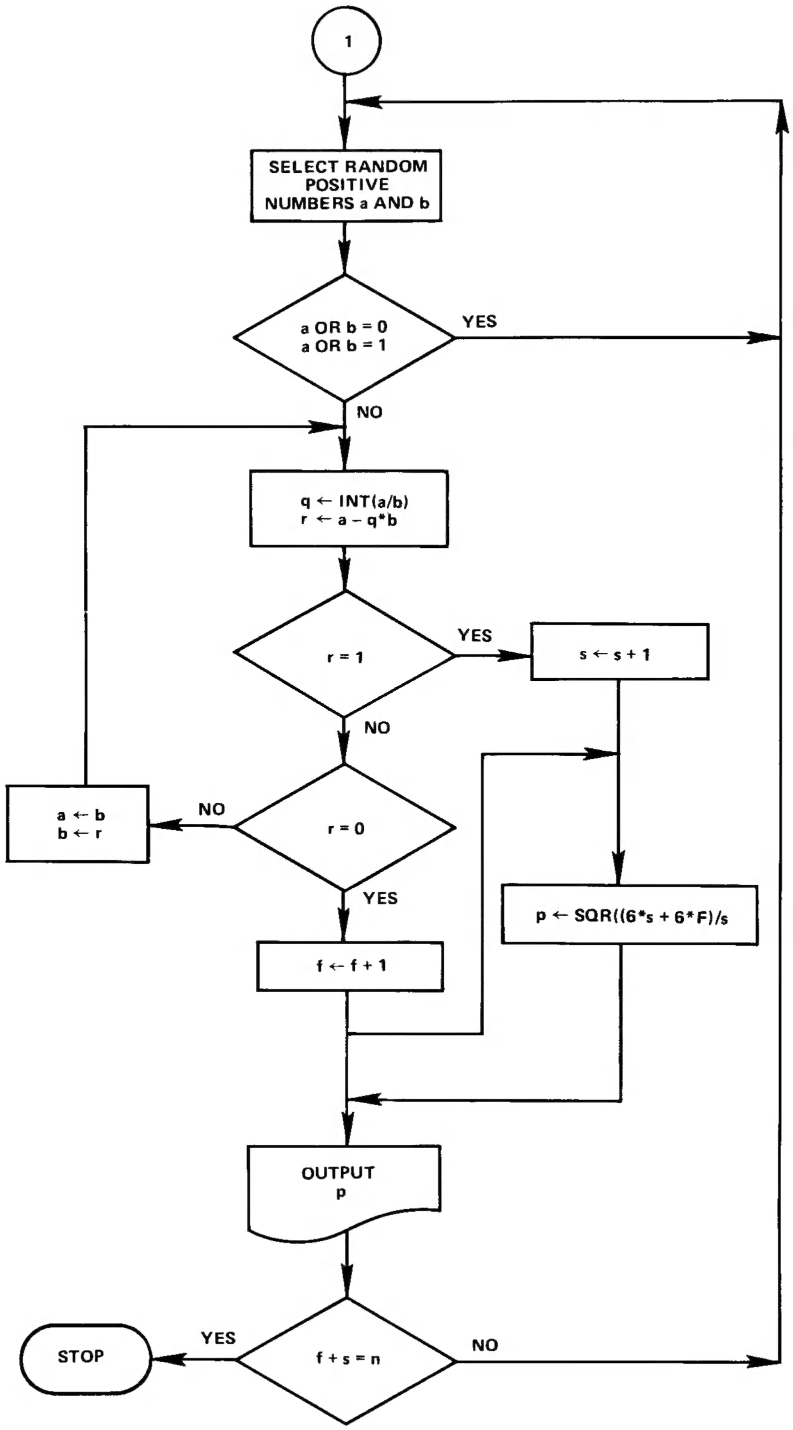
```
RUN

?16,3,5,.001
 5      4      3      2      2.1          2.2          2.3          2.4
 2.5           2.6           2.5          2.51         2.52         2.51
 2.511         2.512         2.513        2.514        2.515        2.516
 2.517         2.518         2.519        2.52         2.519        2.5191
 2.5192        2.5193        2.5194       2.5195       2.5196       2.5197
 3     TH. ROOT OF  16    IS 2.5198

DONE
RUN

?81,4,2,.001
 2      4     TH. ROOT OF  81    IS 3

DONE
RUN

?.0234,2,.01,.00001
 .01           1.01          9.99999E-03    .11           .21           .11
 .12           .13           .14          .15          .16          .15
 .151          .152          .153         .152         .1521        .1522
 .1523         .1524         .1525        .1526        .1527        .1528
 .1529         .153          .1529        .15291       .15292       .15293
 2     TH. ROOT OF  .0234      IS .15294

DONE
```

EXERCISE 12 – Computing π by a Probability Method

The random number–probability approach to computing π, instead of a circle-related approach, is presented here since it ties in well with the students' study of numbers. Encourage your students to have the computer print the approximation to π after the selection of each pair of random numbers. This gives the student the opportunity to see that the successive approximations gradually approach the actual value of π, Different runs of the program will result in approximations sometimes approaching π as a lower bound, and other times approaching π as an upper bound. Consequently, the problem is a good introduction to the concept of a limit.

Micro Flow Chart




1
SELECT RANDOM POSITIVE NUMBERS a AND b
a OR b = 0
a OR b = 1
YES
NO
q ← INT(a/b)
r ← a – q*b
r = 1
YES
s ← s + 1
NO
r = 0
NO
a ← b
b ← r
YES
f ← f + 1
p ← SQR((6*s + 6*F)/s
OUTPUT
p
f + s = n
YES
STOP
NO

Example Program

Exercise 12.

```
10 REM--A PROGRAM TO COMPUTE PI.
20 LET S=0
30 LET F=0
40 INPUT N
50 FOR X=1 TO N
60 LET A=INT(RND(0)*1000)
70 LET B=INT(RND(0)*1000)
80 IF A=0 OR B=0 THEN 200
90 IF A=1 OR B=1 THEN 200
100 LET Q=INT(A/B)
110 LET R=A-Q*B
120 IF R=1 THEN 170
130 IF R=0 THEN 220
140 LET A=B
150 LET B=R
160 GOTO 100
170 LET S=S+1
180 LET P=SQR((6*S+6*F)/S)
190 PRINT P;
200 NEXT X
210 GOTO 60
220 LET F=F+1
230 IF S=0 THEN 200
240 GOTO 180
250 END
```

```
RUN

?20000
 2.44949     3.4641      3      2.82843     3.16228     3.4641
 3.24037     3.09839     3      2.9277      3.0706      3.20714
 3.33809     3.24037     3.3541      3.4641      3.3665      3.4641
 3.55903     3.65148     3.74166     3.63318     3.54196     3.4641
 3.53553     3.60555     3.53009     3.59487     3.52542     3.4641
 3.40955     3.36067     3.41278     3.3665      3.32455     3.28634
 3.25137     3.21926     3.26134     3.30289     3.34392     3.31006
 3.27872     3.24962     3.22252     3.19722     3.23179     3.26599
 3.24037     3.27327     3.24834     3.2249      3.20282     3.2329
 3.21131     3.1909      3.17157     3.19926     3.1803      3.16228
 3.18852     3.21455     3.19628     3.17888     3.16228     3.18651
 3.21056     3.19374     3.21714     3.20061     3.18479     3.20714
 3.22933     3.21334     3.23498     3.25648     3.24037     3.2249
 3.24551     3.26599     3.25042     3.23544     3.25511     3.27466
 3.2596      3.24509     3.2311      3.24962     3.23583     3.25396
 3.27198     3.25813     3.24473     3.26213     3.24893     3.23616
 3.25297     3.2697      3.28634     3.30289     3.28959     3.30584
 3.322       3.33809     3.32455     3.31142     3.29869     3.28634
 3.30152     3.31663     3.33167     3.3191      3.30688     3.29502
 3.28348     3.27226     3.28634     3.27525     3.2891      3.27815
 3.26749     3.28096     3.29438     3.30774     3.32106     3.33431
 3.32317     3.31231     3.30173     3.3145      3.30404     3.29383
 3.28387     3.27414     3.26464     3.25537     3.26731     3.27921
 3.26991     3.26081     3.25192     3.26343     3.2749      3.28634
 3.27731     3.28859     3.29983     3.29079     3.30188     3.29293
 3.28416     3.27558     3.26717     3.25894     3.2695      3.28003
 3.27178     3.26369     3.274       3.26599     3.25813     3.25042
 3.24286     3.25279     3.24529     3.23793     3.24767     3.24037
 3.25        3.24276     3.25228     3.26177     3.27124     3.2639
 3.25669     3.26599     3.27525     3.26803     3.26093     3.25396
 3.24711     3.25606     3.265       3.25813     3.26697     3.26015
 3.25345     3.24685     3.25548     3.26408     3.25747     3.25097
 3.25943     3.25297     3.24662     3.24037     3.23422     3.22817
 3.22221     3.21634     3.22437     3.23238     3.22649     3.23443
```

```
 3.22858      3.22282      3.21714      3.2249       3.21926      3.2137
 3.20822      3.21581      3.21037      3.21789      3.21248      3.20714
 3.20187      3.19668      3.19156      3.18651      3.18153      3.18872
 3.18377      3.17888      3.17406      3.1693       3.1646       3.17157
 3.17851      3.18544      3.19236      3.19926      3.20615      3.20128
 3.19647      3.19172      3.19847      3.20521      3.20045      3.19574
 3.1911       3.19771      3.20431      3.19965      3.19505      3.1905
 3.18601      3.19246      3.19889      3.19438      3.20077      3.19628
 3.19185      3.18748      3.19374      3.2          3.20624      3.20182
 3.20802      3.20362      3.19926      3.19496      3.19071      3.18651
 3.18236      3.18835      3.18422      3.19018      3.19612      3.19197
 3.18787      3.19374      3.18966      3.19549      3.19142      3.19722
 3.19317      3.18916      3.19489      3.20061      3.19659      3.19261
 3.18868      3.18479      3.18093      3.17712      3.17335      3.16962
 3.16593      3.16228      3.15866      3.15508      3.16048      3.15691
 3.15338      3.14989      3.1552       3.15172      3.157        3.16228
 3.15878      3.15532      3.16054      3.16575      3.17095      3.16746
 3.164        3.16914      3.16569      3.17081      3.16737      3.16397
 3.16059      3.16564      3.17068      3.16729      3.1723       3.16893
 3.16559      3.16228      3.16721      3.16392      3.16882      3.17373
 3.17862      3.17529      3.17199      3.16873      3.16549      3.16228
 3.16707      3.17185      3.16863      3.17338      3.17813      3.1749
 3.1717       3.16853      3.16539      3.17006      3.16692      3.17157
 3.16844      3.16535      3.16228      3.15924      3.15622      3.15323
 3.15776      3.15478      3.15182      3.14889      3.14598      3.1431
 3.14024      3.13741      3.14181      3.13898      3.14336      3.14774
 3.1449       3.14926      3.14643      3.15076      3.14794      3.14514
 3.14943      3.14664      3.15091      3.14813      3.15238      3.1496
 3.15383      3.15106      3.15527      3.15948      3.1567       3.15394
 3.15811      3.15537      3.15951      3.16366      3.1609       3.15817
 3.15546      3.15955      3.16364      3.16773      3.1718       3.16906
 3.17312      3.17038      3.16766      3.16496      3.16228      3.16628
 3.16361      3.16759      3.17157      3.16889      3.16623      3.16359
 3.16097      3.16489      3.16228      3.15968      3.15711      3.15456
 3.15202      3.1495       3.147        3.14452      3.14833      3.15214
 3.14965      3.15345      3.15096      3.1485       3.14605      3.14362
 3.14736      3.14494      3.14253      3.14014      3.14384      3.14754
 3.15123      3.14882      3.14643      3.15009      3.14771      3.15135
 3.14897      3.15261      3.15023      3.14787      3.15148      3.15508
 3.15868      3.16228      3.15989      3.15752      3.16109      3.15872
 3.16228      3.15992      3.15757      3.1611       3.16463      3.16228
 3.16579      3.1693       3.16694      3.1646       3.16809      3.16575
 3.16922      3.16689      3.16458      3.16802      3.16571      3.16914
 3.16684      3.16455      3.16228      3.16002      3.15777      3.16115
 3.16453      3.16228      3.16004      3.15782      3.15561      3.15895
STOP
```